

HIVE CASE STUDY

PROBLEM STATEMENT

In this modern era, Tech companies are exploring ways to improve their sales by analysing customer behaviour and gaining insights about product trends. In order to make better business decisions, E-commerce websites are finding their way by tracking the number of clicks made by customers and their spending time on websites in searching for patterns within them. This kind of collected data is called a click stream data. Furthermore, the websites make it easier for customers to find the products they require without much scavenging. In this case study, we are working with click stream data by getting insights and making decisions upon how the E-commerce websites can improve their sales.

OBJECTIVE

The aim is to extract the data and gather insights from a real-life data set of an e-commerce company.

DATA

The data used in this assignment is a public clickstream dataset of a cosmetics store. The clickstream data contains all the logs as to how one navigated through the ecommerce website. It also contains other details such as customer time spent on every page, number of clicks made, adding items to the cart, customer id etc.



Case Study Submitted by: Aditya Kumar & Ruchika Sharma

The data is available from the link provided:

<https://e-commerce-events-ml.s3.amazonaws.com/2019-Oct.csv>

<https://e-commerce-events-ml.s3.amazonaws.com/2019-Nov.csv>

OVERVIEW OF STEPS

- **Copying the data set into the HDFS:**

- Launch an EMR cluster that utilizes the Hive services, and
- Move the data from the S3 bucket into the HDFS

- **Creating the database and launching Hive queries on your EMR cluster:**

- Create the structure of your database,
- Use optimized techniques to run your queries as efficiently as possible
- Show the improvement of the performance after using optimization on any single query.
- Run Hive queries to answer the questions given below.

- **Cleaning up**

- Drop your database, and
- Terminate your cluster

EMR Cluster Creation

EMR Cluster Landing Page > Create Cluster > Advanced Options > Selecting the release emr-5.29 and the required services.

Create Cluster - Advanced Options [Go to quick options](#)

Step 1: Software and Steps <ul style="list-style-type: none"> Step 2: Hardware Step 3: General Cluster Settings Step 4: Security 	<p>Software Configuration</p> <p>Release: emr-5.29.0</p> <table border="0" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; vertical-align: top;"> <input checked="" type="checkbox"/> Hadoop 2.8.5 <input type="checkbox"/> JupyterHub 1.0.0 <input type="checkbox"/> Ganglia 3.7.2 <input checked="" type="checkbox"/> Hive 2.3.6 <input type="checkbox"/> MXNet 1.5.1 <input checked="" type="checkbox"/> Hue 4.4.0 <input checked="" type="checkbox"/> Spark 2.4.4 </td> <td style="width: 50%; vertical-align: top;"> <input type="checkbox"/> Zeppelin 0.8.2 <input type="checkbox"/> Tez 0.9.2 <input type="checkbox"/> HBase 1.4.10 <input type="checkbox"/> Presto 0.227 <input type="checkbox"/> Sqoop 1.4.7 <input type="checkbox"/> Phoenix 4.14.3 <input type="checkbox"/> HCatalog 2.3.6 </td> <td style="width: 50%; vertical-align: top;"> <input type="checkbox"/> Livy 0.6.0 <input type="checkbox"/> Flink 1.9.1 <input checked="" type="checkbox"/> Pig 0.17.0 <input type="checkbox"/> ZooKeeper 3.4.14 <input type="checkbox"/> Mahout 0.13.0 <input type="checkbox"/> Oozie 5.1.0 <input type="checkbox"/> TensorFlow 1.14.0 </td> </tr> </table>	<input checked="" type="checkbox"/> Hadoop 2.8.5 <input type="checkbox"/> JupyterHub 1.0.0 <input type="checkbox"/> Ganglia 3.7.2 <input checked="" type="checkbox"/> Hive 2.3.6 <input type="checkbox"/> MXNet 1.5.1 <input checked="" type="checkbox"/> Hue 4.4.0 <input checked="" type="checkbox"/> Spark 2.4.4	<input type="checkbox"/> Zeppelin 0.8.2 <input type="checkbox"/> Tez 0.9.2 <input type="checkbox"/> HBase 1.4.10 <input type="checkbox"/> Presto 0.227 <input type="checkbox"/> Sqoop 1.4.7 <input type="checkbox"/> Phoenix 4.14.3 <input type="checkbox"/> HCatalog 2.3.6	<input type="checkbox"/> Livy 0.6.0 <input type="checkbox"/> Flink 1.9.1 <input checked="" type="checkbox"/> Pig 0.17.0 <input type="checkbox"/> ZooKeeper 3.4.14 <input type="checkbox"/> Mahout 0.13.0 <input type="checkbox"/> Oozie 5.1.0 <input type="checkbox"/> TensorFlow 1.14.0
<input checked="" type="checkbox"/> Hadoop 2.8.5 <input type="checkbox"/> JupyterHub 1.0.0 <input type="checkbox"/> Ganglia 3.7.2 <input checked="" type="checkbox"/> Hive 2.3.6 <input type="checkbox"/> MXNet 1.5.1 <input checked="" type="checkbox"/> Hue 4.4.0 <input checked="" type="checkbox"/> Spark 2.4.4	<input type="checkbox"/> Zeppelin 0.8.2 <input type="checkbox"/> Tez 0.9.2 <input type="checkbox"/> HBase 1.4.10 <input type="checkbox"/> Presto 0.227 <input type="checkbox"/> Sqoop 1.4.7 <input type="checkbox"/> Phoenix 4.14.3 <input type="checkbox"/> HCatalog 2.3.6	<input type="checkbox"/> Livy 0.6.0 <input type="checkbox"/> Flink 1.9.1 <input checked="" type="checkbox"/> Pig 0.17.0 <input type="checkbox"/> ZooKeeper 3.4.14 <input type="checkbox"/> Mahout 0.13.0 <input type="checkbox"/> Oozie 5.1.0 <input type="checkbox"/> TensorFlow 1.14.0		

Hardware Configuration Page > To define the cluster & nodes : Instance type for both master & core nodes are M4.large

Node type	Instance type	Instance count	Purchasing option
Master Master - 1	m4.large	1 Instances	<input checked="" type="radio"/> On-demand <input type="radio"/> Spot <input type="checkbox"/> Use on-demand as max price
Core Core - 2	m4.large	1 Instances	<input checked="" type="radio"/> On-demand <input type="radio"/> Spot <input type="checkbox"/> Use on-demand as max price

Case Study Submitted by: Aditya Kumar & Ruchika Sharma

Naming the cluster uniquely.

General Options

Cluster name CaseStudy_Hive

Logging [i](#)
S3 folder s3://aws-logs-620094501754-us-east-1/elasticmapred

Debugging [i](#)

Termination protection [i](#)

Selecting the key-pair (created before creating the cluster)

Security Options

EC2 key pair Hive_Assignment [i](#)

Cluster visible to all IAM users in account [i](#)

Permissions [i](#)

Default Custom
Use default IAM roles. If roles are not present, they will be automatically created for you with managed policies for automatic policy updates.

EMR role EMR_DefaultRole [i](#)

EC2 instance profile EMR_EC2_DefaultRole [i](#)

Auto Scaling role EMR_AutoScaling_DefaultRole [i](#)

Cluster “CaseStudy_Hive ” is successfully created and launched.

Clusters						
Actions		Name	ID	Status	Creation time (UTC+2)	Elapsed time
Create cluster	View details	Clone	Terminate	CaseStudy_Hive	j-1W738NHRUH3ZR <small>Waiting Cluster ready</small>	2021-07-02 13:17 (UTC+2) 18 minutes 0

“Hive ” is the Key-Pair created for this case study.

Key pairs (5) [Info](#)

[Filter key pairs](#)

<input type="checkbox"/>	Name	Fingerprint	ID
<input type="checkbox"/>	airline_test_key	cf:1d:b2:77:15:de:00:cd:a1:87:e1:a5:e0...	key-085d45793b6534195
<input type="checkbox"/>	Hive_Assignment	0f:4e:6f:c8:12:62:1f:f6:7c:5a:33:38:19:...	key-07309743827c5baa5

Case Study Submitted by: Aditya Kumar & Ruchika Sharma

❖ Hadoop & Hive Queries :

Terminal > Connecting to EMR Cluster using ssh.

```
(base) pratyushachillarige@MacBook-Pro ~ % cd Desktop/Keys
(base) pratyushachillarige@MacBook-Pro Keys % chmod 400 Hive_assignment.pem
(base) pratyushachillarige@MacBook-Pro Keys % ssh -i Hive_assignment.pem hadoop@ec2-3-238-98-119.compute-1.amazonaws.com
The authenticity of host 'ec2-3-238-98-119.compute-1.amazonaws.com (3.238.98.119)' can't be established.
ECDSA key fingerprint is SHA256:4cIAS8cBAGWJYIso0lPfRDbb8cHKr0IMC+8p+ZGnJky4.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-238-98-119.compute-1.amazonaws.com,3.238.98.119' (ECDSA) to the list of known hosts.
```

```
--| --|_ )
-| ( _ /   Amazon Linux AMI
---|\_\_|_||

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
56 package(s) needed for security, out of 102 available
Run "sudo yum update" to apply all updates.
■bash: warning: setlocale: LC_CTYPE: cannot change locale (UTF-8): No such file or directory

EEEEEEEEEEEEEEEEEE MMMMMMM          MMMMMMM RRRRRRRRRRRRRR
E:::::::-----: E M:::::M       M:::::M R:::::-----:R
EE:::::EEEEE:::::E M:::::M       M:::::M R::::RRRRR:::R
 E::::E     EEEE M:::::M       M:::::M RR:::R      R:::::R
 E:::E     M::::M:::M       M:::M:::M R:::R      R:::::R
 E:::::EEEEE:::::E M::::M M::::M M:::::M       R::::RRRRR:::R
 E:::::::::::E M::::M M::::M:::M M:::::M       R:::::::::::R
 E:::::EEEEE:::::E M::::M M:::::M       M:::::M R:::::RRRRR:::R
 E::::E     EEEE M::::M       M:::::M R:::::R      R:::::R
 E::::E     EEEE M::::M       MMM  M:::::M R:::::R      R:::::R
EE:::::EEEEE:::::E M::::M       M:::::M R:::::R      R:::::R
E:::::::-----: E M::::M       M:::::M RR:::R      R:::::R
EEEEEEEEEEEEEEEEEE MMMMMMM          MMMMMMM RRRRRRRR      RRRRRR
```

Creating a directory “casestudy”

```
hadoop fs -mkdir /casestudy
```

```
hadoop fs -ls /
```

```
drwxr-xr-x  -  hdfs  hadoop    0 2021-07-06 20:20 /
drwxr-xr-x  -  hadoop hadoop    0 2021-07-06 20:20 /
drwxrwxrwt -  hdfs  hadoop    0 2021-07-06 20:20 /
drwxr-xr-x  -  hdfs  hadoop    0 2021-07-06 20:20 /
drwxr-xr-x  -  hdfs  hadoop    0 2021-07-06 20:20 /
```

```
[hadoop@ip-172-31-66-163 ~]$ hadoop fs -mkdir /casestudy
```

```
[hadoop@ip-172-31-66-163 ~]$ hadoop fs -ls /
```

```
Found 5 items
```

```
drwxr-xr-x  -  hdfs  hadoop    0 2021-07-06 20:20 /
drwxr-xr-x  -  hadoop hadoop    0 2021-07-06 20:20 /
drwxrwxrwt -  hdfs  hadoop    0 2021-07-06 20:20 /
drwxr-xr-x  -  hdfs  hadoop    0 2021-07-06 20:20 /
drwxr-xr-x  -  hdfs  hadoop    0 2021-07-06 20:20 /
```

```
[hadoop@ip-172-31-66-163 ~]$ hadoop fs -mkdir /casestudy
```

```
[hadoop@ip-172-31-66-163 ~]$ hadoop fs -ls /
```

```
Found 5 items
```

Loading the datasets into HDFS from S3:

```
hadoop distcp 's3://e-commerce-events-ml/2019-Oct.csv' /casestudy/2019_Oct.csv
```

Case Study Submitted by: Aditya Kumar & Ruchika Sharma

```
[hadoop@ip-172-31-66-163 ~]$ hadoop distcp 's3://e-commerce-events-ml/2019-Oct.csv' /casestudy/2019_Oct.csv
21/07/03 20:37:03 INFO tools.DistCp: Input Options: DistCpOptions{atomicCommit=false, syncFolder=false, deleteMissing=false, ignoreFailures=false, overwrite=false, skipCRC=false, blocking=true, numListstatusThreads=0, maxMaps=20, mapBandwidth=100, ss lConfigurationFile='null', copyStrategy='uniformsize', preserveStatus=[], preserveRawXattrs=false, atomicWorkPath=null, logPa th=null, sourceFileListing=null, sourcePaths=[s3://e-commerce-events-ml/2019-Oct.csv], targetPath=/casestudy/2019_Oct.csv, ta rgetPathExists=false, filtersFile='null'}
```

```
File Output Format Counters
  Bytes Written=0
DistCp Counters
  Bytes Copied=482542278
  Bytes Expected=482542278
  Files Copied=1
```

hadoop distcp 's3://e-commerce-events-ml/2019-Nov.csv' /casestudy/2019_Nov.csv

```
[hadoop@ip-172-31-66-163 ~]$ hadoop distcp 's3://e-commerce-events-ml/2019-Nov.csv' /casestudy/2019_Nov.csv
21/07/03 20:39:47 INFO tools.DistCp: Input Options: DistCpOptions{atomicCommit=false, syncFolder=false, deleteMissing=false, ignoreFailures=false, overwrite=false, skipCRC=false, blocking=true, numListstatusThreads=0, maxMaps=20, mapBandwidth=100, ss lConfigurationFile='null', copyStrategy='uniformsize', preserveStatus=[], preserveRawXattrs=false, atomicWorkPath=null, logPa th=null, sourceFileListing=null, sourcePaths=[s3://e-commerce-events-ml/2019-Nov.csv], targetPath=/casestudy/2019_Nov.csv, ta rgetPathExists=false, filtersFile='null'}
```

```
File Output Format Counters
  Bytes Written=0
DistCp Counters
  Bytes Copied=545839412
  Bytes Expected=545839412
  Files Copied=1
```

Viewing the data

hadoop fs -cat /casestudy/2019_Oct.csv | head

```
[hadoop@ip-172-31-66-163 ~]$ hadoop fs -cat /casestudy/2019_Oct.csv | head
event_time,event_type,product_id,category_id,category_code,brand,price,user_id,user_session
2019-10-01 00:00:00 UTC,cart,5773203,1487580005134238553,,runail,2.62,463240011,26dd6e6e-4dac-4778-8d2c-92e149dab885
2019-10-01 00:00:03 UTC,cart,5773353,1487580005134238553,,runail,2.62,463240011,26dd6e6e-4dac-4778-8d2c-92e149dab885
2019-10-01 00:00:07 UTC,cart,5881589,2151191071051219817,,lovely,13.48,429681830,49e8d843-adf3-428b-a2c3-fe8bc6a307c9
2019-10-01 00:00:07 UTC,cart,5723490,1487580005134238553,,runail,2.62,463240011,26dd6e6e-4dac-4778-8d2c-92e149dab885
2019-10-01 00:00:15 UTC,cart,5881449,1487580013522845895,,lovely,0.56,429681830,49e8d843-adf3-428b-a2c3-fe8bc6a307c9
2019-10-01 00:00:16 UTC,cart,5857269,1487580005134238553,,runail,2.62,430174032,73dea1e7-664e-43f4-8b30-d32b9d5af04f
2019-10-01 00:00:19 UTC,cart,5739055,1487580008246412266,,kapous,4.75,377667011,81326ac6-daa4-4f0a-b488-fd0956a78733
2019-10-01 00:00:24 UTC,cart,5825598,1487580009445982239,,0.56,467916806,2f5b5546-b8cb-9ee7-7ecd-84276f8ef486
2019-10-01 00:00:25 UTC,cart,5698989,1487580006317032337,,,1.27,385985999,d30965e8-1101-44ab-b45d-cc1bb9fae694
```

hadoop fs -cat /casestudy/2019_Nov.csv | head

```
[hadoop@ip-172-31-66-163 ~]$ hadoop fs -cat /casestudy/2019_Nov.csv | head
event_time,event_type,product_id,category_id,category_code,brand,price,user_id,user_session
2019-11-01 00:00:02 UTC,view,5802432,1487580009286598681,,,0.32,562076640,09fafd6c-6c99-46b1-834f-33527f4de241
2019-11-01 00:00:09 UTC,cart,5844397,1487580006317032337,,,2.38,553329724,2067216c-31b5-455d-a1cc-af0575a34ffb
2019-11-01 00:00:10 UTC,view,5837166,1783999064103190764,,pnb,22.22,556138645,57ed222e-a54a-4907-9944-5a875c2d7f4f
2019-11-01 00:00:11 UTC,cart,5876812,1487580010100293687,,jessnail,3.16,564506666,186c1951-8052-4b37-adce-dd9644b1d5f7
2019-11-01 00:00:24 UTC,remove_from_cart,5826182,1487580007483048900,,,3.33,553329724,2067216c-31b5-455d-a1cc-af0575a34ffb
2019-11-01 00:00:24 UTC,remove_from_cart,5826182,1487580007483048900,,,3.33,553329724,2067216c-31b5-455d-a1cc-af0575a34ffb
2019-11-01 00:00:25 UTC,view,5856189,1487580009026551821,,runail,15.71,562076640,09fafd6c-6c99-46b1-834f-33527f4de241
2019-11-01 00:00:32 UTC,view,5837835,1933472286753424063,,,3.49,514649199,432a4e95-375c-4b40-bd36-0fc039e77580
2019-11-01 00:00:34 UTC,remove_from_cart,5870838,1487580007675986893,,milv,0.79,429913900,2f0bff3c-252f-4fe6-afcd-5d8a6a92839a
```

Datasets are successfully loaded.

Launch Hive

```
[hadoop@ip-172-31-66-163 ~]$ hive
Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.properties Async: false
hive> show databases ;
OK
default
Time taken: 0.697 seconds, Fetched: 1 row(s)
```

Creating new database “hive_assignment”

```
hive> CREATE DATABASE IF NOT EXISTS hive_assignment ;
hive> SHOW DATABASES ;
hive> DESCRIBE DATABASE hive_assignment ;
```

```
hive> CREATE DATABASE IF NOT EXISTS hive_assignment ;
OK
Time taken: 0.316 seconds
hive> SHOW DATABASES ;
OK
default
hive_assignment
Time taken: 0.021 seconds, Fetched: 2 row(s)
hive> DESCRIBE DATABASE hive_assignment ;
OK
hive_assignment          hdfs://ip-172-31-66-163.ec2.internal:8020/user/hive/warehouse/hive_assignment.db      hadoop  USER
Time taken: 0.046 seconds, Fetched: 1 row(s)
```

Creating new table “retail”

```
hive > CREATE EXTERNAL TABLE IF NOT EXISTS retail (event_time timestamp, event_type string,
product_id string, category_id string, category_code string, brand string, price decimal(10,3), user_id bigint,
user_session string) ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde' WITH
SERDEPROPERTIES ("separatorChar" = ",", "quoteChar" = "\"", "escapeChar" = "\\") stored as textfile
LOCATION '/casestudy' TBLPROPERTIES ("skip.header.line.count"="1") ;
```

```
hive> CREATE EXTERNAL TABLE IF NOT EXISTS retail (event_time timestamp, event_type string, product_id string, category_id string, category_code string, brand string, price decimal(10,3), user_id bigint, user_session string) ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
WITH SERDEPROPERTIES ("separatorChar" = ",", "quoteChar" = "\"", "escapeChar" = "\\") stored as textfile LOCATION '/casestudy' TBLPROPERTIES (
"skip.header.line.count"="1") ;
OK
Time taken: 0.33 seconds
```

```
hive> DESCRIBE retail ;
hive> DESCRIBE retail ;
OK
event_time          string          from deserializer
event_type          string          from deserializer
product_id          string          from deserializer
category_id         string          from deserializer
category_code       string          from deserializer
brand               string          from deserializer
price               string          from deserializer
user_id              string          from deserializer
user_session        string          from deserializer
Time taken: 0.107 seconds, Fetched: 9 row(s)
```

Loading data into table “retail”;

```
hive> LOAD DATA INPATH '/casestudy/2019_Oct.csv' INTO TABLE retail ;
hive> LOAD DATA INPATH '/casestudy/2019_Nov.csv' INTO TABLE retail;
hive> LOAD DATA INPATH '/casestudy/2019_Oct.csv' INTO TABLE retail ;
Loading data to table default.retail
OK
Time taken: 1.353 seconds
hive> LOAD DATA INPATH '/casestudy/2019_Nov.csv' INTO TABLE retail;
Loading data to table default.retail
OK
Time taken: 0.673 seconds
```

Performing data check:

```
hive> SELECT * FROM retail WHERE MONTH(event_time)=11 limit 5 ;
hive> SELECT * FROM retail WHERE MONTH(event_time)=10 limit 5 ;
```

```
hive> SELECT * FROM retail WHERE MONTH(event_time)=11 limit 5 ;
OK
2019-11-01 00:00:02 UTC view    5802432 1487580009286598681          0.32   562076640      09fafd6c-6c99-46b1-834f-33527f4de241
2019-11-01 00:00:09 UTC cart    5844397 1487580006317032337          2.38   553329724      2067216c-31b5-455d-a1cc-af0575a34ff
2019-11-01 00:00:10 UTC view    5837166 1783999864183190764          pnb    22.22   556138645      57ed222e-a54a-4907-9944-5a875c2d7f4f
2019-11-01 00:00:11 UTC cart    5876812 1487580010100293687          jessnail   3.16   564506666      186c1951-8052-4b37-adce-dd9644b1d5f7
2019-11-01 00:00:24 UTC remove_from_cart 5826182 148758000748304890          3.33   553329724      2067216c-31b5-455d-a1cc-af0575a34ff
Time taken: 0.209 seconds, Fetched: 5 row(s)
hive> SELECT * FROM retail WHERE MONTH(event_time)=10 limit 5 ;
OK
2019-10-01 00:00:00 UTC cart    5773283 1487580005134238553          runail  2.62   463240011      26dd6e6e-4dac-4778-8d2c-92e149dab885
2019-10-01 00:00:03 UTC cart    5773353 1487580005134238553          runail  2.62   463240011      26dd6e6e-4dac-4778-8d2c-92e149dab885
2019-10-01 00:00:07 UTC cart    5881589 2151191071051219817          lovely   13.48   429681830      49e8d843-adf3-428b-a2c3-fe8bbc6a307c9
2019-10-01 00:00:07 UTC cart    5723498 1487580005134238553          runail  2.62   463240011      26dd6e6e-4dac-4778-8d2c-92e149dab885
2019-10-01 00:00:15 UTC cart    5881449 1487580013522845895          lovely   0.56   429681830      49e8d843-adf3-428b-a2c3-fe8bbc6a307c9
Time taken: 0.195 seconds, Fetched: 5 row(s)
```

QUESTION 1:

Find the total revenue generated due to purchases made in October.

```
hive> SELECT SUM(price) FROM retail WHERE MONTH(event_time)=10 AND event_type='purchase' ;
hive> SELECT SUM(price) FROM retail WHERE MONTH(event_time)=10 AND event_type='purchase' ;
Query ID = hadoop_20210703205507_ed5f7747-bed8-43fe-8928-96a4c51de12e
Total jobs = 1
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.
Status: Running (Executing on YARN cluster with App id application_1625344162494_0004)

-----

| VERTICES        | MODE      | STATUS    | TOTAL | COMPLETED | RUNNING | PENDING | FAILED | KILLED |
|-----------------|-----------|-----------|-------|-----------|---------|---------|--------|--------|
| Map 1 .....     | container | SUCCEEDED | 2     | 2         | 0       | 0       | 0      | 0      |
| Reducer 2 ..... | container | SUCCEEDED | 1     | 1         | 0       | 0       | 0      | 0      |


-----  

VERTICES: 02/02 [=====>>] 100% ELAPSED TIME: 61.27 s
-----  

OK  

1211538.4299997438  

Time taken: 70.514 seconds, Fetched: 1 row(s)
```

Time Taken to execute the above query is 70.5 sec.

This is very high. Hence, to reduce this execution time, we will dynamically partition the table “retail” and add buckets to create an optimised table.

DYNAMIC PARTITIONING

```
hive> set hive.exec.dynamic.partition=true;
hive> set hive.exec.dynamic.partition.mode=nonstrict;
```

```
[hive> set hive.exec.dynamic.partition=true;
[hive> set hive.exec.dynamic.partition.mode=nonstrict;
```

PARTITION TABLE 1: retail_part_1

Partition on : event_type (there are 4 types and all questions are related to ‘purchase’)

```
hive> CREATE EXTERNAL TABLE IF NOT EXISTS retail_part_1 (event_time timestamp, product_id string, category_id string, category_code string, brand string, price decimal(10,3), user_id bigint, user_session string) PARTITIONED BY(event_type string) CLUSTERED BY (user_id) INTO 5 buckets ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde' STORED AS textfile ;
```

```
hive> DESCRIBE retail_part_1;
```

```
hive> CREATE EXTERNAL TABLE IF NOT EXISTS retail_part_1 (event_time timestamp, product_id string, category_id string, category_code string, brand string, price decimal(10,3), user_id bigint, user_session string) PARTITIONED BY(event_type string) CLUSTERED BY (user_id) INTO 5 buckets ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde' STORED AS textfile ;
OK
Time taken: 0.091 seconds
hive> DESCRIBE retail_part_1
>;
OK
event_time          string          from deserializer
product_id          string          from deserializer
category_id          string          from deserializer
category_code        string          from deserializer
brand               string          from deserializer
price               decimal(10,3)   from deserializer
user_id              bigint         from deserializer
user_session         string          from deserializer
event_type           string          from deserializer

# Partition Information
# col_name          data_type      comment
event_type           string
Time taken: 0.11 seconds, Fetched: 14 row(s)
```

```
hive> INSERT INTO TABLE retail_part_1 PARTITION (event_type) SELECT event_time, product_id, category_id, category_code, brand, price, user_id, user_session, event_type FROM retail ;
```

```
hive> INSERT INTO TABLE retail_part_1 PARTITION (event_type) SELECT event_time, product_id, category_id, category_code, brand, price, user_id, user_session, event_type FROM retail ;
Query ID = hadoop_20210703210049_7f21892f-38a2-4a94-936b-e5832bf3e4c2
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1625344162494_0004)

-----
 VERTICES    MODE      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
Map 1 ..... container  SUCCEEDED   2      2        0        0        0        0
Reducer 2 .... container  SUCCEEDED   5      5        0        0        0        0
-----
VERTICES: 02/02  [=====>>>] 100%  ELAPSED TIME: 157.65 s

Loading data to table default.retail_part_1 partition (event_type=null)

Loaded : 4/4 partitions.
          Time taken to load dynamic partitions: 0.396 seconds
          Time taken for adding to write entity : 0.002 seconds
OK
Time taken: 159.703 seconds
```

Executing the same query with the new table “retail_part_1” to check the time.

```
hive> SELECT SUM(price) FROM retail_part_1 WHERE MONTH(event_time)=10 AND event_type='purchase' ;
```

```
[hive> SELECT SUM(price) FROM retail_part_1 WHERE MONTH(event_time)=10 AND event_type='purchase' ;
Query ID = hadoop_20210703210459_b8fc7436-88e8-45ae-a5a6-ed362fbc8db1
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1625344162494_0004)

-----
      VERTICES      MODE      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
-----
Map 1 ..... container  SUCCEEDED    3        3        0        0        0        0        0
Reducer 2 ..... container  SUCCEEDED    1        1        0        0        0        0        0
-----
VERTICES: 02/02  [=====>>>] 100%  ELAPSED TIME: 24.26 s
-----
OK
1211538.4299998982
Time taken: 25.365 seconds, Fetched: 1 row(s)
```

Time Taken to execute the above query is 25.36 sec.

PARTITION TABLE 2: retail_part_3

Partition on : month

```
hive> CREATE EXTERNAL TABLE IF NOT EXISTS retail_part_3 (event_time timestamp, event_type string, product_id string, category_id string, category_code string, brand string, price decimal(10,3), user_id bigint, user_session string) PARTITIONED BY(month int) CLUSTERED BY (brand) INTO 5 buckets ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde' STORED AS textfile ;
```

```
hive> DESCRIBE retail_part_3 ;
```

```
hive> CREATE EXTERNAL TABLE IF NOT EXISTS retail_part_3 (event_time timestamp, event_type string, product_id string, category_id string, category_code string, brand string, price decimal(10,3), user_id bigint, user_session string) PARTITIONED BY(month int) CLUSTERED BY (brand) INTO 5 buckets ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde' STORED AS textfile ;
OK
Time taken: 0.096 seconds
hive> DESCRIBE retail_part_3
> ;
OK
event_time          string          from deserializer
event_type          string          from deserializer
product_id          string          from deserializer
category_id         string          from deserializer
category_code       string          from deserializer
brand               string          from deserializer
price               decimal(10,3)   from deserializer
user_id              bigint         from deserializer
user_session        string          from deserializer
month                int            from deserializer
# Partition Information
# col_name           data_type      comment
month                int
Time taken: 0.046 seconds, Fetched: 15 row(s)
```

```
hive> INSERT INTO TABLE retail_part_3 PARTITION (month) SELECT event_time, event_type, product_id, category_id, category_code, brand, price, user_id, user_session, MONTH(CAST(REPLACE(event_time,'UTC','') AS timestamp)) FROM retail ;
```

```

hive> INSERT INTO TABLE retail_part_3 PARTITION (month) SELECT event_time, event_type, product_id, category_id, category_code, brand, price,
user_id, user_session, MONTH(CAST(REPLACE(event_time,'UTC','') AS timestamp)) FROM retail ;
Query ID = hadoop_20210703214135_edc02768-9163-475d-8361-ca592e0ea982
Total jobs = 1
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.
Status: Running (Executing on YARN cluster with App id application_1625344162494_0005)

-----  

      VERTICES    MODE     STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED  

-----  

Map 1 ..... container  SUCCEEDED   2       2       0       0       0       0  

Reducer 2 ..... container  SUCCEEDED   5       5       0       0       0       0  

-----  

VERTICES: 02/02  [=====>] 100%  ELAPSED TIME: 182.32 s  

-----  

Loading data to table default.retail_part_3 partition (month=null)  

  

Loaded : 2/2 partitions.  

Time taken to load dynamic partitions: 0.137 seconds  

Time taken for adding to write entity : 0.0 seconds  

OK  

Time taken: 191.306 seconds

```

Executing the same query with the new table “retail_part_3” to check the time.

```

hive> SELECT SUM(price) FROM retail_part_3 WHERE MONTH(event_time)=10 AND
event_type='purchase';

```

```

[hive> SELECT SUM(price) FROM retail_part_3 WHERE MONTH(event_time)=10 AND event_type='purchase' ;
Query ID = hadoop_20210703214854_e7370545-f651-4db1-8459-f900bc422d4c
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1625344162494_0005)

-----  

      VERTICES    MODE     STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED  

-----  

Map 1 ..... container  SUCCEEDED   8       8       0       0       0       0  

Reducer 2 ..... container  SUCCEEDED   1       1       0       0       0       0  

-----  

VERTICES: 02/02  [=====>] 100%  ELAPSED TIME: 77.27 s  

-----  

OK  

1211538.4299999713  

Time taken: 77.853 seconds, Fetched: 1 row(s)

```

Time Taken to execute the above query is 77.85 sec.

We get an optimised table by Partitioning on “event_type” and clustering by “user_id”. Hence, for all the following analysis, we will be using the optimised table “retail_part_1”.

QUESTION 1:

Find the total revenue generated due to purchases made in October.

```
hive> SELECT SUM(price) FROM retail_part_1 WHERE MONTH(event_time)=10 AND event_type='purchase';
```

```
hive> SELECT SUM(price) FROM retail_part_1 WHERE MONTH(event_time)=10 AND event_type='purchase' ;
Query ID = hadoop_20210703210459_b8fc7436-88e8-45ae-a5a6-ed362fbc8db1
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1625344162494_0004)

-----
      VERTICES      MODE      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
-----
Map 1 ..... container  SUCCEEDED    3        3        0        0        0        0        0
Reducer 2 ..... container  SUCCEEDED    1        1        0        0        0        0        0
-----
VERTICES: 02/02  [=====>>>] 100%  ELAPSED TIME: 24.26 s
-----
OK
1211538.4299998982
Time taken: 25.365 seconds, Fetched: 1 row(s)
```

Time Taken to execute the above query is 25.36 sec.

QUESTION 2:

Write a query to yield the total sum of purchases per month in a single output.

```
hive> SELECT MONTH(event_time), SUM(price) as sum_purchase, COUNT(event_type) as cnt FROM retail_part_1 WHERE event_type='purchase' GROUP BY MONTH(event_time);
```

```
hive> SELECT MONTH(event_time), SUM(price) as sum_purchase, COUNT(event_type) as cnt FROM retail_part_1 WHERE event_type='purchase'
GROUP BY MONTH(event_time) ;
Query ID = hadoop_20210703210616_7fc0c1ca-76a8-4348-b534-f8d12e1a161d
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1625344162494_0004)

-----
      VERTICES      MODE      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
-----
Map 1 ..... container  SUCCEEDED    3        3        0        0        0        0        0
Reducer 2 ..... container  SUCCEEDED    1        1        0        0        0        0        0
-----
VERTICES: 02/02  [=====>>>] 100%  ELAPSED TIME: 23.37 s
-----
OK
10      1211538.4299998982      245624
11      1531016.8999999384      322417
Time taken: 24.33 seconds, Fetched: 2 row(s)
```

In October month, 245624 purchases generated revenue of 1211538. Similarly in November month, 322417 purchases generated revenue of 1531016.89

QUESTION 3:

Write a query to find the change in revenue generated due to purchases from October to November.

```
hive> WITH diff AS ( SELECT SUM(CASE WHEN date_format(event_time,'MM')=10 THEN price ELSE 0 END) AS October, SUM(CASE WHEN date_format(event_time,'MM')=11 THEN price ELSE 0 END) AS November FROM retail_part_1 WHERE date_format(event_time,'MM') IN (10,11) AND event_type='purchase') SELECT October, November, (November - October) as Difference FROM diff ;
```

```
hive> WITH diff AS ( SELECT SUM(CASE WHEN date_format(event_time,'MM')=10 THEN price ELSE 0 END) AS October, SUM(CASE WHEN date_format(event_time,'MM')=11 THEN price ELSE 0 END) AS November FROM retail_part_1 WHERE date_format(event_time,'MM') IN (10,11) AND event_type='purchase') SELECT October, November, (November - October) as Difference FROM diff ;
Query ID = hadoop_20210703210953_a38a14aa-84c8-41a5-ae6e-bf0888c0084e
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1625344162494_0004)

-----  

      VERTICES    MODE     STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED  

-----  

Map 1 ..... container    SUCCEEDED      3        3        0        0        0        0  

Reducer 2 ..... container    SUCCEEDED      1        1        0        0        0        0  

-----  

VERTICES: 02/02  [=====>>>] 100%  ELAPSED TIME: 38.57 s  

-----  

OK  

1211538.429999898     1531016.8999999384     319478.4700000405  

Time taken: 39.19 seconds, Fetched: 1 row(s)
```

The change in revenue generated from October to November is 319478.47

QUESTION 4:

Find distinct categories of products. Categories with null category code can be ignored.

```
hive> SELECT DISTINCT split(category_code,'\\.')[0] AS category FROM retail_part_1 WHERE split(category_code,'\\.')[0]<>" ;
```

```
hive> SELECT DISTINCT split(category_code,'\\.')[0] AS category FROM retail_part_1 WHERE split(category_code,'\\.')[0]<>" ;
Query ID = hadoop_20210703211146_29be0e65-1d20-4e3a-b38c-02df54611237
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1625344162494_0004)

-----  

      VERTICES    MODE     STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED  

-----  

Map 1 ..... container    SUCCEEDED      6        6        0        0        0        0  

Reducer 2 ..... container    SUCCEEDED      5        5        0        0        0        0  

-----  

VERTICES: 02/02  [=====>>>] 100%  ELAPSED TIME: 66.68 s  

-----  

OK  

furniture  

appliances  

accessories  

apparel  

sport  

stationery
```

There are 6 distinct categories of products. They are: Furniture, appliances, accessories, apparel, sport and stationary.

QUESTION 5:

Find the total number of products available under each category.

```
hive> SELECT split(category_code,'\\.')[0] AS category, COUNT(product_id) AS prd FROM retail_part_1  
GROUP BY split(category_code,'\\.')[0] ORDER BY prd DESC ;
```

```
hive> SELECT split(category_code,'\\.')[0] AS category, COUNT(product_id) AS prd FROM retail_part_1 GROUP BY split  
(category_code,'\\.')[0] ORDER BY prd DESC ;  
Query ID = hadoop_20210703211417_e390ef09-668a-47dc-be35-f21d8eb9ebfb  
Total jobs = 1  
Launching Job 1 out of 1  
Status: Running (Executing on YARN cluster with App id application_1625344162494_0004)  
  
-----  
 VERTICES    MODE      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED  
-----  
Map 1 ..... container  SUCCEEDED   6       6        0        0        0        0  
Reducer 2 ..... container  SUCCEEDED   5       5        0        0        0        0  
Reducer 3 ..... container  SUCCEEDED   1       1        0        0        0        0  
-----  
VERTICES: 03/03  [=====>>>] 100%  ELAPSED TIME: 71.77 s  
-----  
OK  
     8594895  
appliances      61736  
stationery       26722  
furniture        23604  
apparel          18232  
accessories       12929  
sport             2  
Time taken: 72.472 seconds, Fetched: 7 row(s)
```

“Sport” category has the least number of products, whereas “appliances” has 61736 products.

QUESTION 6:

Which brand had the maximum sales in October and November combined?

```
SELECT brand, SUM(price) AS Sales FROM retail_part_1 WHERE brand <>'' AND event_type='purchase'  
GROUP BY brand ORDER BY Sales DESC LIMIT 1 ;
```

```
hive> SELECT brand, SUM(price) AS Sales FROM retail_part_1 WHERE brand <>'' AND event_type='purchase' GROUP BY brand  
ORDER BY Sales DESC LIMIT 1 ;  
Query ID = hadoop_20210703211729_b13c2a75-929c-4df6-a5dd-433eb654c55f  
Total jobs = 1  
Launching Job 1 out of 1  
Status: Running (Executing on YARN cluster with App id application_1625344162494_0004)  
  
-----  
 VERTICES    MODE      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED  
-----  
Map 1 ..... container  SUCCEEDED   3       3        0        0        0        0  
Reducer 2 ..... container  SUCCEEDED   1       1        0        0        0        0  
Reducer 3 ..... container  SUCCEEDED   1       1        0        0        0        0  
-----  
VERTICES: 03/03  [=====>>>] 100%  ELAPSED TIME: 21.99 s  
-----  
OK  
runail  148297.93999999898  
Time taken: 22.623 seconds, Fetched: 1 row(s)
```

Brand “runail” has the maximum sales for both months combined.

QUESTION 7:

Which brands increased their sales from October to November?

```
hive>WITH monthly_diff AS ( SELECT brand, SUM(CASE WHEN date_format(event_time,'MM')=10 THEN price ELSE 0 END) AS October, SUM(CASE WHEN date_format(event_time,'MM')=11 THEN price ELSE 0 END) AS November FROM retail_part_1 WHERE event_type='purchase' GROUP BY brand) SELECT brand, October, November, (November-October) as Sales_diff FROM monthly_diff WHERE (November-October) >0 ORDER BY Sales_diff ;
```

```
hive> WITH monthly_diff AS ( SELECT brand, SUM(CASE WHEN date_format(event_time,'MM')=10 THEN price ELSE 0 END) AS October, SUM(CASE WHEN date_format(event_time,'MM')=11 THEN price ELSE 0 END) AS November FROM retail_part_1 WHERE event_type='purchase' GROUP BY brand) SELECT brand, October, November, (November-October) as Sales_diff FROM monthly_diff WHERE (November-October) >0 ORDER BY Sales_diff ;
[Query ID = hadoop_20210703212211_f102df0a-d3aa-49cb-b1d6-721a62ddd4ea]
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1625344162494_0004)

-----  

      VERTICES      MODE      STATUS TOTAL COMPLETED RUNNING PENDING FAILED KILLED  

-----  

Map 1 ..... container SUCCEEDED 3 3 0 0 0 0  

Reducer 2 ..... container SUCCEEDED 1 1 0 0 0 0  

Reducer 3 ..... container SUCCEEDED 1 1 0 0 0 0  

-----  

VERTICES: 03/03 [=====>>>] 100% ELAPSED TIME: 33.93 s  

-----  

OK  

ovalie 2.54 3.1 0.56  

cosima 20.22999999999997 20.93 0.70000000000028  

grace 100.9199999999999 102.6100000000001 1.690000000000261  

helloganic 0.0 3.1 3.1  

skinity 8.88 12.44000000000001 3.560000000000005  

bodyton 1376.3400000000006 1380.640000000003 4.29999999999727  

moyou 5.71 10.2800000000001 4.570000000000001  

neoleor 43.41 51.7 8.29000000000006
```

beautix	10493.94999999984	12222.94999999992	1729.000000000073
milv	3904.939999999796	5642.009999999875	1737.070000000008
masura	31266.079999998827	33058.469999995	1792.3900000007234
f.o.x	6624.22999999996	8577.27999999995	1953.049999999993
kapous	11927.159999999969	14093.080000000014	2165.9200000000455
concept	11832.13999999992	13380.39999999941	2348.259999999493
estel	21756.74999999916	24142.67000000035	2385.9200000001183
kaypro	881.34 3268.69999999994	2387.35999999999	
benovy	409.619999999999	3259.97000000001	2850.3500000000013
italwax	21940.23999999896	24799.36999999915	2859.130000000019
yoko	8756.91 11707.87999999986	2950.969999999866	
haruyama	9390.68999999879	12352.91000000069	2962.2200000001903
marathon	7280.7500000003	10273.89999999999	2992.34999999996
lovely	8704.37999999994	11939.0599999998	3234.679999999857
bpw.style	11572.15000000003	14837.44000000017	3265.2900000000864
staleks	8519.73000000014	11875.61000000015	3355.88000000001
freedecor	3421.77999999996	7671.79999999959	4250.01999999963
runail	71539.27999999 76758.6599999984	5219.380000000849	
polarus	6013.71999999999	11371.93000000004	5358.21000000005
cosmoprofi	8322.80999999994	14536.99000000042	6214.18000000048
jessnail	26287.84000000127	33345.23000000014	7057.39000000014
strong	29196.63000000005	38671.2700000002	9474.64000000014
ingarden	23161.38999999883	33566.21000000225	10404.82000000342
lianail	5892.839999999865	16394.2399999996	10501.39999999976
uno	35302.03000000006	51039.75000000007	15737.72000000067
grattol	35445.5399999993	71472.71000000341	36027.17000000348
	474679.060000175	619509.2400000119	144830.1799999435

Total of 161 brands have increased their sales from October to November.

QUESTION 8:

Your company wants to reward the top 10 users of its website with a Golden Customer plan.
Write a query to generate a list of top 10 users who spend the most.

```
hive>SELECT user_id, SUM(price) AS expense FROM retail_part_1 WHERE event_type='purchase'  
GROUP BY user_id ORDER BY expense DESC LIMIT 10 ;
```

```
hive> SELECT user_id, SUM(price) AS expense FROM retail_part_1 WHERE event_type='purchase' GROUP BY user_id  
ORDER BY expense DESC LIMIT 10 ;  
Query ID = hadoop_20210703212639_400e03e8-b947-4da4-839e-8ae7020c6264  
Total jobs = 1  
Launching Job 1 out of 1  
Status: Running (Executing on YARN cluster with App id application_1625344162494_0004)  
  
-----  
 VERTICES      MODE      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED  
-----  
Map 1 ..... container  SUCCEEDED   3       3        0        0        0        0  
Reducer 2 ..... container  SUCCEEDED   1       1        0        0        0        0  
Reducer 3 ..... container  SUCCEEDED   1       1        0        0        0        0  
-----  
VERTICES: 03/03  [=====>>>] 100%  ELAPSED TIME: 23.20 s  
-----  
OK  
557790271      2715.869999999991  
150318419      1645.970000000005  
562167663      1352.85  
531900924      1329.45  
557850743      1295.480000000005  
522130011      1185.389999999999  
561592095      1109.700000000005  
431950134      1097.589999999997  
566576008      1056.360000000004  
521347209      1040.909999999999  
Time taken: 23.839 seconds, Fetched: 10 row(s)
```

Above is the list of the top 10 users (User_ids alongwith the amount spent) who spend the most.

❖ Cleaning Up:

Once the analysis is completed, deleting the database & terminating the cluster

```
[hive> SHOW DATABASES ;
OK
default
hive_assignment
Time taken: 0.012 seconds, Fetched: 2 row(s)
[hive> DROP DATABASE hive_assignment ;
OK
Time taken: 0.204 seconds
[hive> SHOW DATABASES ;
OK
default
Time taken: 0.009 seconds, Fetched: 1 row(s)
```

Cluster Overview			
		Name	ID
		Status	
<input type="checkbox"/>	▶	CaseStudy_Hive	j-3SEPFQE5FK1RT Terminated User request